

dLab 1 - Introdução a Linguagem Java

Neste laboratório faremos Instalação do **Java Development Kit**, em ambiente Windows ou Linux. E escreveremos nosso primeiro programa em Java, o mais famoso programa do mundo.

Exercícios

Exercício 1: Instalar JDK 7.0 (30 minutos)

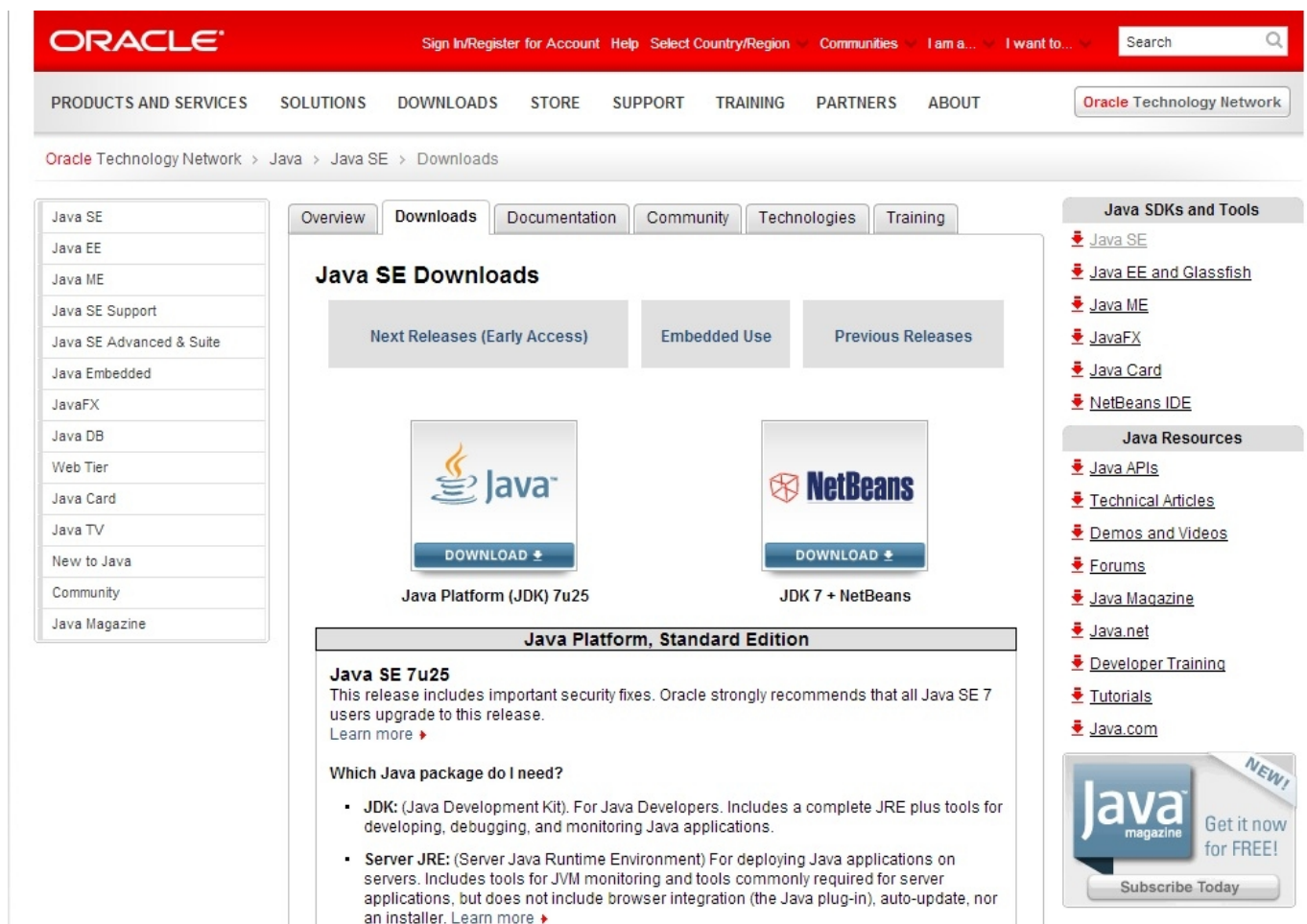
Exercício 2: Escrever, compilar e rodar um programa **HelloWorld** em Java usando ferramentas da linha de comando (30 minutos)

Exercício 1 - Instalar JDK 7

1.1 - Fazendo Download do JDK 7

Para você baixar o JDK(Java Development Kit), acesse o link no site da ORACLE:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

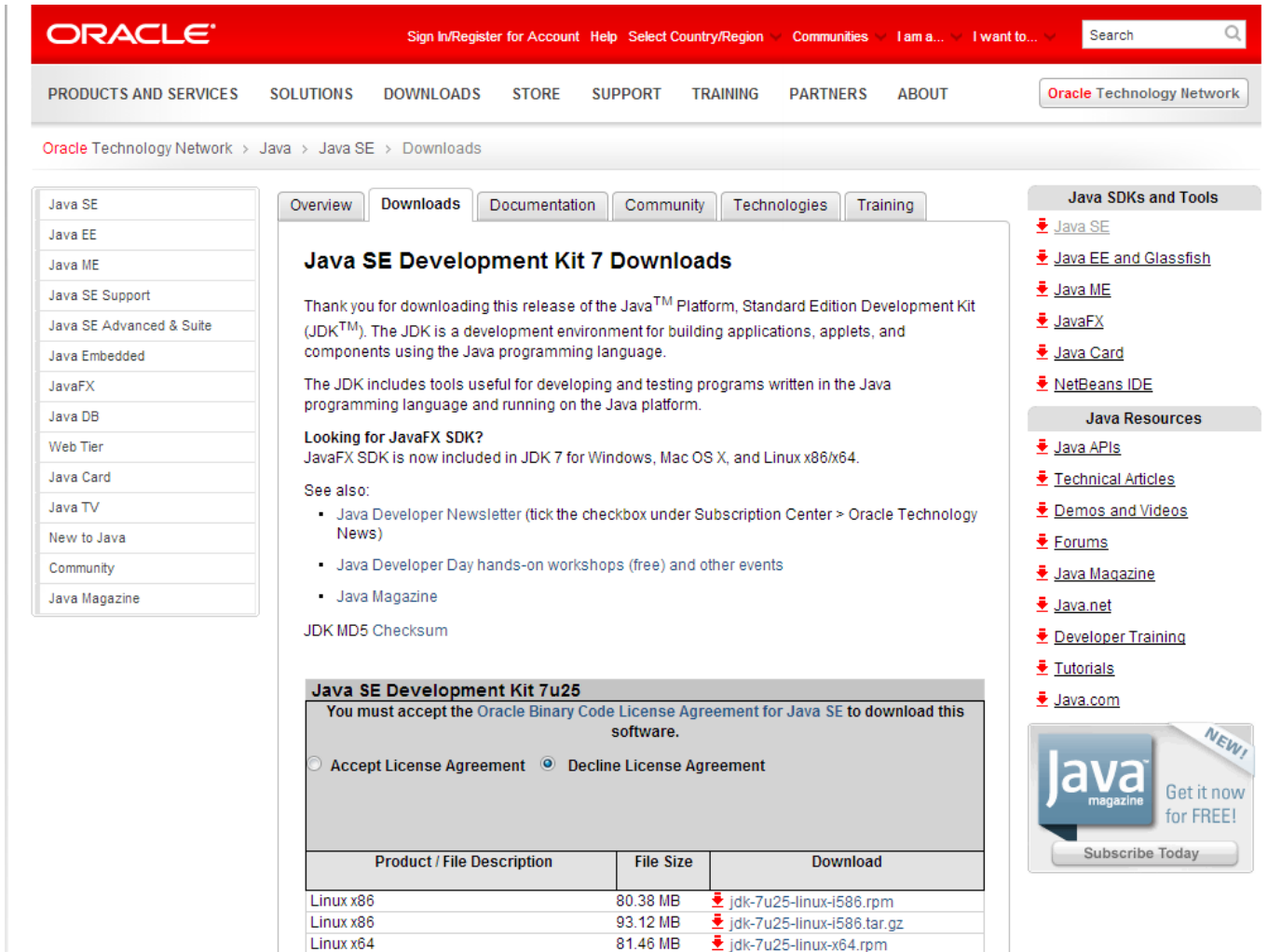


The screenshot shows the Oracle Java SE Downloads page. The header includes the Oracle logo and navigation links like 'Sign In/Register for Account', 'Help', 'Select Country/Region', 'Communities', 'I am a...', and 'I want to...'. A search bar is also present. The main navigation bar lists 'PRODUCTS AND SERVICES', 'SOLUTIONS', 'DOWNLOADS', 'STORE', 'SUPPORT', 'TRAINING', 'PARTNERS', and 'ABOUT'. The breadcrumb trail reads 'Oracle Technology Network > Java > Java SE > Downloads'. The left sidebar contains a list of links: Java SE, Java EE, Java ME, Java SE Support, Java SE Advanced & Suite, Java Embedded, JavaFX, Java DB, Web Tier, Java Card, Java TV, New to Java, Community, and Java Magazine. The main content area is titled 'Java SE Downloads' and has tabs for 'Overview', 'Downloads', 'Documentation', 'Community', 'Technologies', and 'Training'. Under 'Downloads', there are three sections: 'Next Releases (Early Access)', 'Embedded Use', and 'Previous Releases'. The 'Previous Releases' section features two download buttons: 'DOWNLOAD' for 'Java Platform (JDK) 7u25' and 'DOWNLOAD' for 'JDK 7 + NetBeans'. Below these, a section titled 'Java Platform, Standard Edition' provides details about 'Java SE 7u25', stating it includes important security fixes and recommending an upgrade. It also includes a 'Which Java package do I need?' section with two options: 'JDK: (Java Development Kit)' and 'Server JRE: (Server Java Runtime Environment)'. The right sidebar lists 'Java SDKs and Tools' (Java SE, Java EE and Glassfish, Java ME, JavaFX, Java Card, NetBeans IDE) and 'Java Resources' (Java APIs, Technical Articles, Demos and Videos, Forums, Java Magazine, Java.net, Developer Training, Tutorials, Java.com). At the bottom right, there is a 'Java Magazine' subscription banner with a 'Subscribe Today' button.

Nesta página, você tem a opção de baixar o **JDK, JRE, Documentação, Código Fonte** e

outras.

1. Clique em **Download (Java Platform JDK 7u25)** como mostra a figura acima.
2. Você será redirecionado para a página abaixo.



Oracle Sign In/Register for Account Help Select Country/Region Communities I am a... I want to... Search

PRODUCTS AND SERVICES SOLUTIONS DOWNLOADS STORE SUPPORT TRAINING PARTNERS ABOUT Oracle Technology Network

Oracle Technology Network > Java > Java SE > Downloads

Overview Downloads Documentation Community Technologies Training

Java SE Development Kit 7 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

Looking for JavaFX SDK?
JavaFX SDK is now included in JDK 7 for Windows, Mac OS X, and Linux x86/x64.

See also:

- Java Developer Newsletter (tick the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK MD5 Checksum

Java SE Development Kit 7u25

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☐ Accept License Agreement ☒ Decline License Agreement

Product / File Description	File Size	Download
Linux x86	80.38 MB	jdk-7u25-linux-i586.rpm
Linux x86	93.12 MB	jdk-7u25-linux-i586.tar.gz
Linux x64	81.46 MB	jdk-7u25-linux-x64.rpm

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [JavaFX](#)
- [Java Card](#)
- [NetBeans IDE](#)

Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Java.net](#)
- [Developer Training](#)
- [Tutorials](#)
- [Java.com](#)

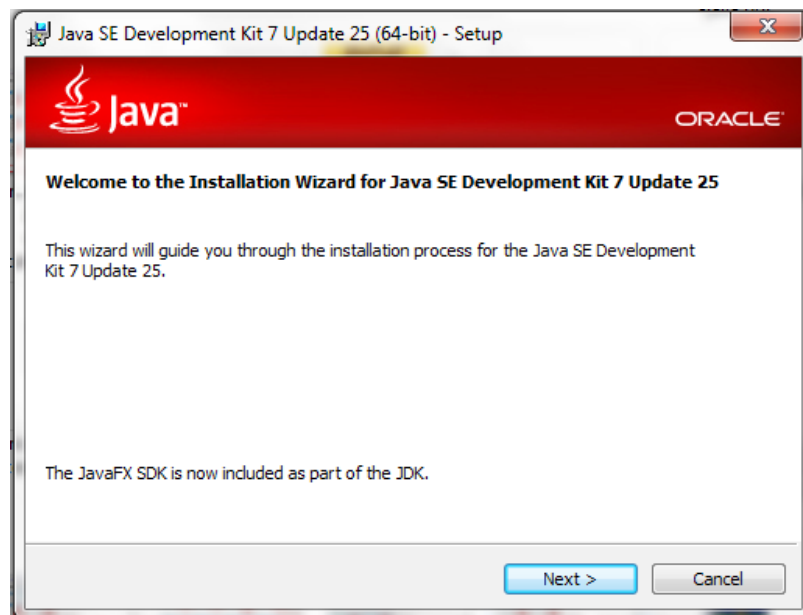
NEW!
Java magazine
Get it now for FREE!
Subscribe Today

1. Habilite o download aceitando os termos da licença. **Accept License Agreement**
2. Escolha sua plataforma SO: **Windows** para plataforma **Windows**, por exemplo, e **click** em **Download**.
3. Selecione sua plataforma SO na coluna de Download, para iniciar o download

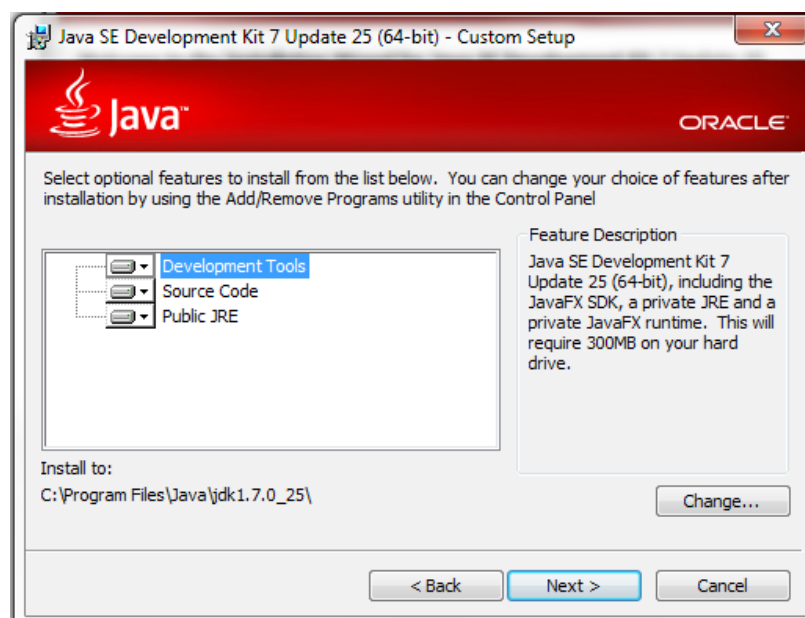
1.2 - Instalação do JDK em ambiente Windows

Para instalar o JDK no Windows, primeiro baixe-o no site da Oracle, é um simples arquivo executável que contém o Wizard de instalação.

1. Dê um clique duplo no arquivo **jdk-<versão>-windows-<arquitetura>.exe**, e espere até ele entrar no wizard de instalação.
2. Nesta tela Clique em **Next**.

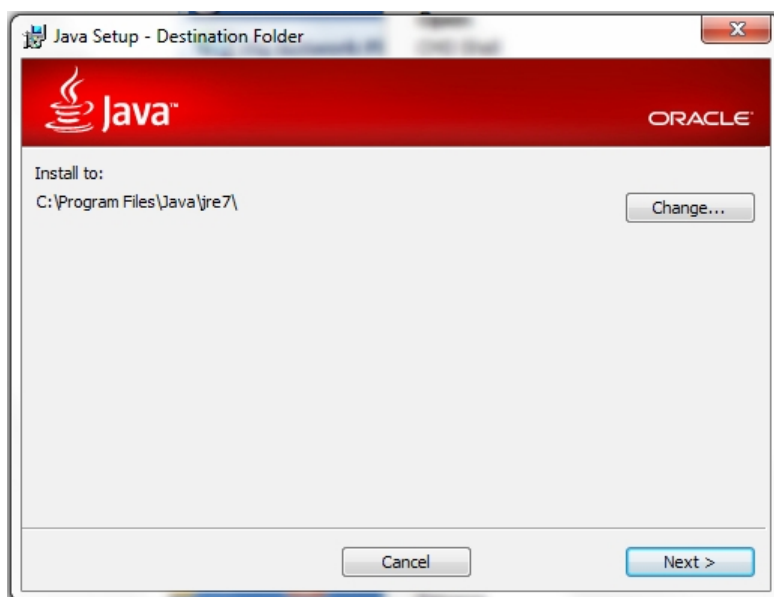


3. Agora devemos selecionar quais recursos instalaremos junto com o java (Ferramentas de desenvolvimento, o código fonte e o próprio java), e onde ele será instalado (marque esse caminho porque usaremos ele mais pra frente), deixe como está e clique em **Next**.

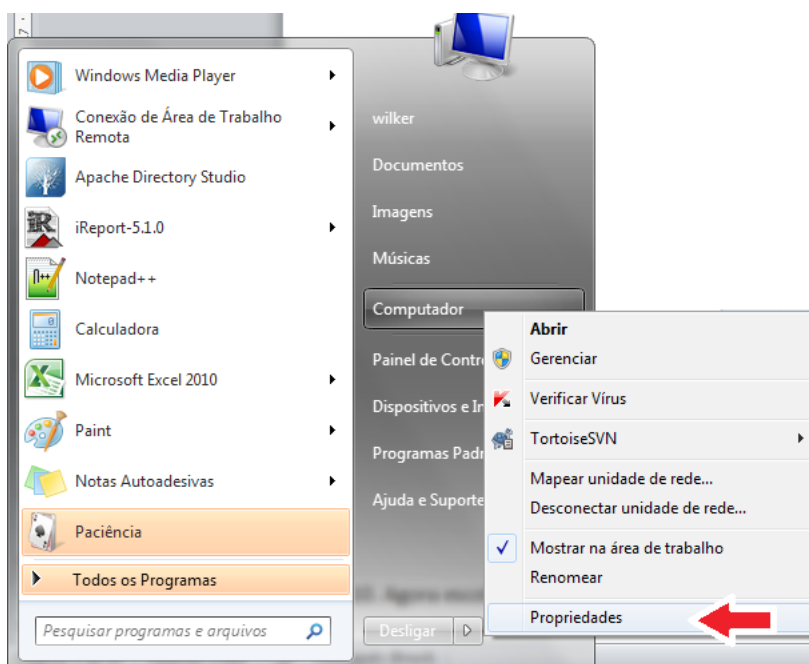


4. O processo de instalação iniciará.

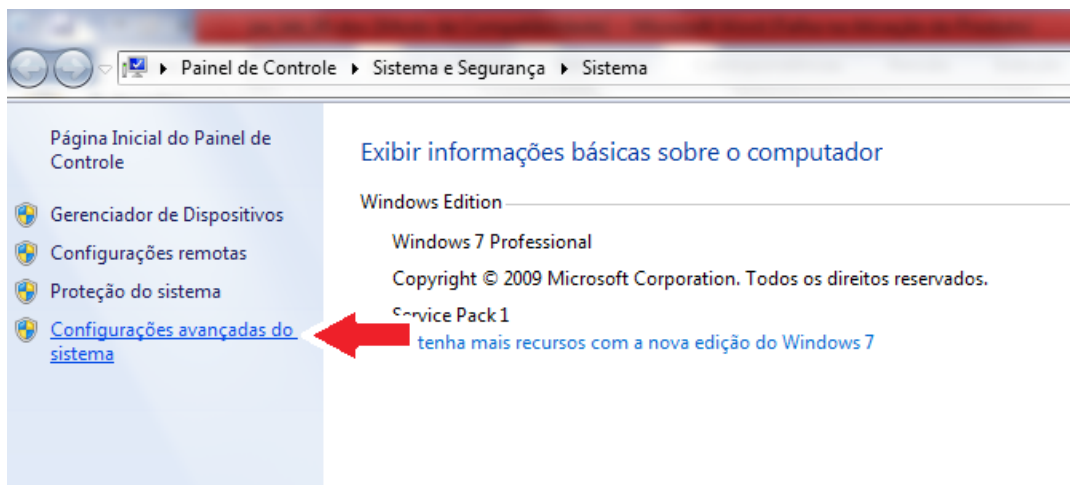
5. Este passo só será executado caso você ainda não tenha instalado o JRE na sua máquina. Agora ele começará a instalar o **JRE(Java Runtime Environment)**. Assim como o **JDK**, ele também tem algumas opções. Deixe como está e clique em **Next**.



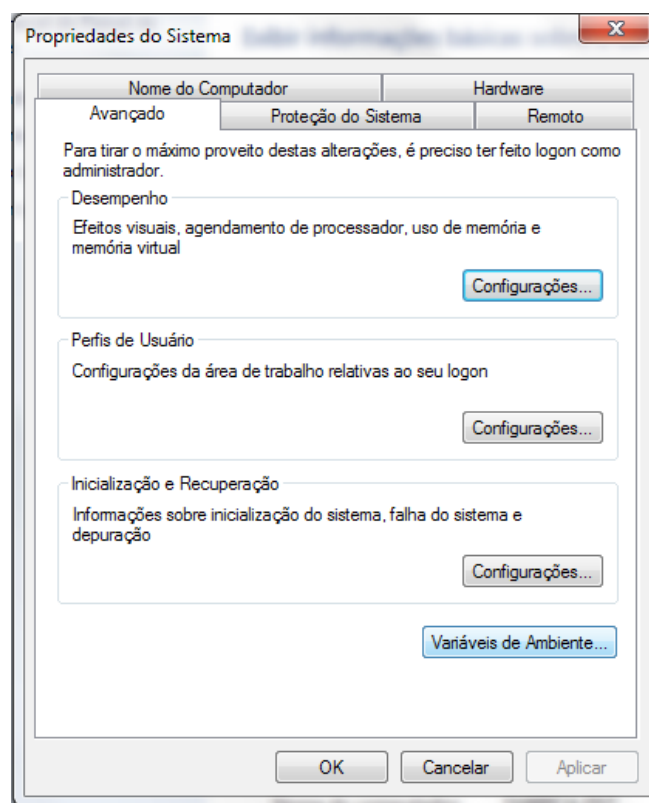
6. O processo de instalação da **JRE(Java Runtime Environment)** iniciará.
7. Agora seu **JDK** está instalado. Clique em **Finish**.
8. Agora vamos criar as **Variáveis de Ambiente**. Clique com o botão direito em cima do ícone **Meu Computador** e selecione a opção **Propriedades**.



9. Clique em **Configurações avançadas do sistema**.

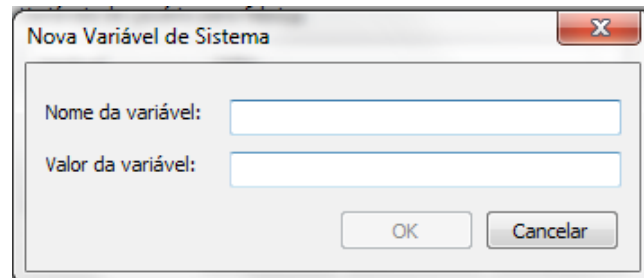


10. Agora escolha a aba **Avançado** e depois clique no botão **Variáveis de Ambiente**.

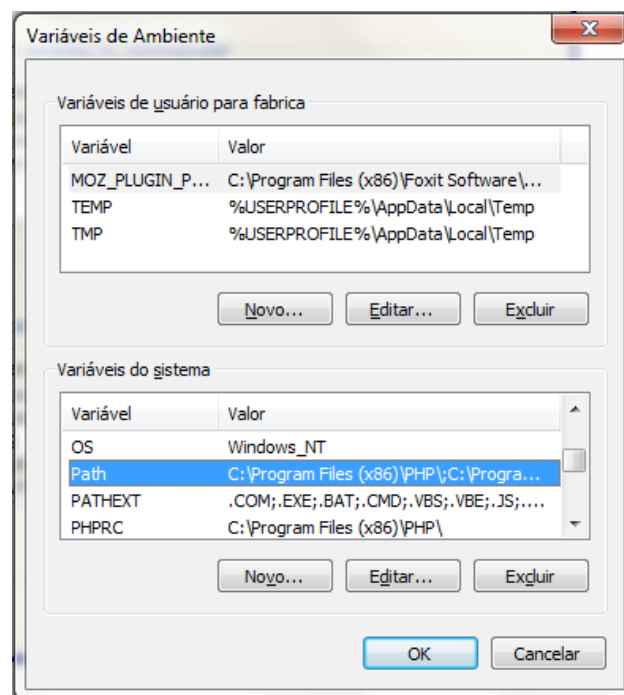


11. Nesta tela você verá na parte de cima, as variáveis de ambiente do usuário corrente, e embaixo, as variáveis de ambiente do computador, que serve para todos os usuários. Clique no botão **Novo** da parte de baixo.

12. Agora em **Nome da Variável** digite **JAVA_HOME** e em **Valor da Variável** digite o **Caminho** que você anotou no passo 3, ou seja, o caminho que foi instalado o JDK. Provavelmente vai estar em **C:\Program Files\Java\jdk1.7.0_25**, ou um caminho parecido com este dependendo da versão que foi instalado. Depois clique em **OK**.



13. Agora não vamos criar outra variável e sim alterar, para isso procure a variável **PATH**, ou **Path**, e clique no botão **Editar**.



14. Não altere o nome da variável e não apague os dados nela informados! Deixe como está e adicione no final do valor **;%JAVA_HOME%\bin**, não esqueça do ponto-e-vírgula no início, assim você está adicionando mais um caminho à sua variável **Path**.

15. Agora abra o **prompt** e digite **javac -version** para mostrar a versão do **Java Compiler** e algumas opções, caso não apareça reveja os passos e confira se não esqueceu ou pulou nenhum deles e se o caminho colocado nas variáveis estão corretos.

1.3 - Instalação do JDK em ambiente Linux

1. Baixe o pacote de instalação para Linux chamado **Linux self-extracting file**, assim o arquivo que você baixará terá a extensão **.bin**.
2. Abra um terminal e faça login como **root** (comando **su**).
3. Crie um diretório chamado **/java** na raiz do seu sistema (**mkdir /java**)
4. Copie o arquivo que você baixou para o diretório **/java** (**cp /<caminho do arquivo>/jdk-<versão>-linux.bin /java**)
5. Execute o arquivo (**sh jdk<versão>-linux.bin**)
6. Aparecerá na sua tela um acordo de instalação, aperte a tecla **Q** para ir até o final do documento e depois escreva **yes** para aceitar o contrato. Feito isso ele começará a descompactar o JDK em um diretório que ele cria, chamado **jdk<versão>**.
7. Vamos agora criar as variáveis de ambiente. Edite o arquivo. Usando um editor de sua preferência, altere seu arquivo **.bashrc** ou **.bash_profile** (depende da sua distribuição e sua preferência) ou ainda **/etc/bash.bashrc**, considerando que você não tem privilégios de super usuário, faça:

```
> ~/bashrc  
> vi ~/bashrc
```

Adicione as seguintes linhas no final do arquivo, substituindo **<versão>** pela versão que foi instalado do JDK:

```
JAVA_HOME=/java/jdk<versão>  
PATH=$PATH:/$JAVA_HOME/bin  
CLASSPATH=.export JAVA_HOME PATH CLASSPATH
```

8. Execute o comando **source ~/.bashrc**
10. Abra um terminal e teste o **Java Compiler** executando o comando **javac -version**
11. A saída deverá ser parecida com a apresentada abaixo, se não aparecer reveja os passos e confira se foi feito tudo corretamente, inclusive o caminho dos diretórios.

```
javac 1.7.0_25  
javac: no source files  
Usage: javac <options> <source files>  
where possible options include:  
.....
```


Exercício 2 -. Escrever, compilar e executar programa “Hello World” usando ferramentas de linha de comando

O objetivo do exercício é promover uma experiência completa como um ciclo de desenvolvimento de programas em Java – escrever, compilar e executar um programa java tão simples quanto possível – usando as ferramentas de linha de comando do seu JDK. Se você já programou em linguagens como C, C++ ou Pascal (o tradicional, não a IDE Delphi), verá que há pequenas diferenças.

Em Java o compilador que é chamado **javac** compila o arquivo de código fonte Java em um arquivo binário chamado de **bytecode** que pode ser executado em qualquer plataforma compatível com Java, isto provê a portabilidade dos programas em Java. Os **bytecodes** são os arquivos ***.java** compilados para serem executados ,se transformando nos arquivos com extensão ***.class** .

1. Montar e executar o programa Java **HelloWorld**, usando o comando **javac** e **java**
2. Usar o compilador Java (comando: **javac**)
3. Usar o interpretador Java (comando: **java**)

2.1 Montar e executar o programa Java HelloWorld usando o comando *javac* e *java*

1. Crie um diretório onde você vai colocar os programas em Java que você escrever

```
>mkdir c:\myjavaprograms
```

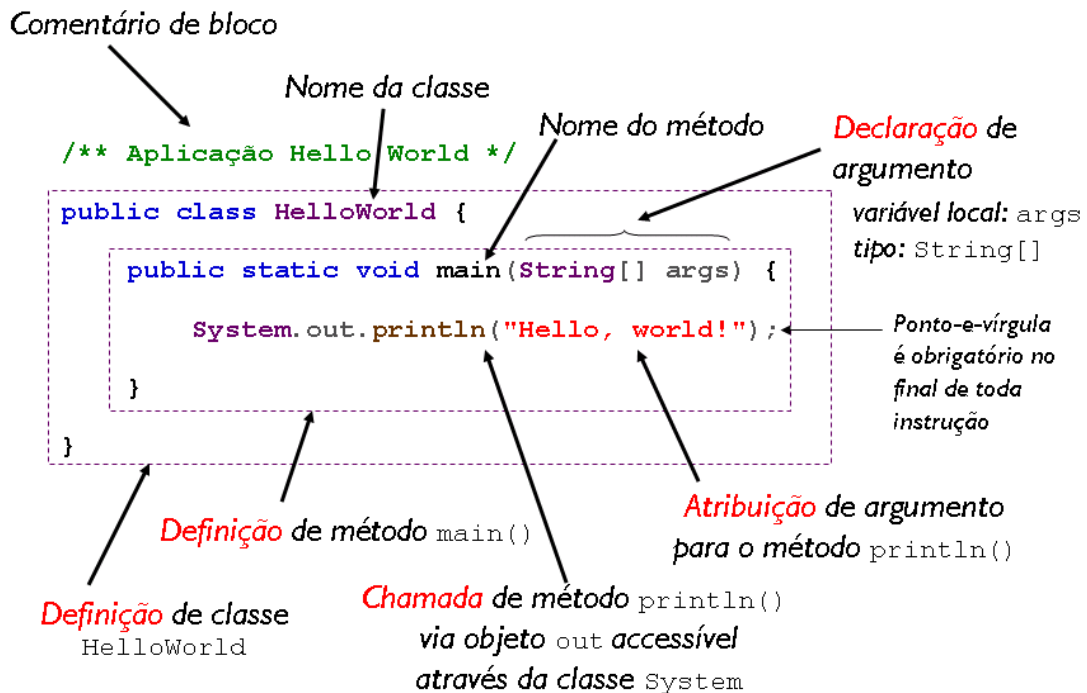
```
>cd \myjavaprograms
```

2. Escreva o código **HelloWorld.java** usando o editor de sua preferência, pode ser **textpad** ou **notepad** no Windows; **vi** ou **gedit** no Linux. Veja o gabarito do programa na **Listagem-0.1** abaixo. Nestes primeiros exemplos é interessante que você digite e tenha experiência com alguns erros compilação.

```
public class HelloWorld {  
  
    /**  
    *Aplicação Hello World  
    */  
    public static void main( String[] args ){  
  
        // Imprime cadeia de caracteres "Hello world" na tela  
        System.out.println("Hello, world");  
    }  
}
```

Listagem -0.1: HelloWorld.java

Anatomia do Código:



3. Compile **HelloWorld.java** usando o compilador **javac**. O compilador **javac** foi instalado com seu **J2SE SDK**. Ele está no diretório **%JAVA_HOME%\bin** (Windows) ou **\$JAVA_HOME/bin** (Linux). O resultado da compilação será a criação do arquivo **HelloWorld.class**.

> **javac HelloWorld.java**

Problema #1: se você receber a mensagem de erro abaixo

> **HelloWorld.java**

'javac' is not recognized as an internal or external command, operable program or batch file

Isto significa que **%JAVA_HOME%\bin** para Windows ou **\$JAVA_HOME/bin** para Linux, não está configurada corretamente. Você pode tentar colocando **C:\Program Files\Java\jdk1.7.0_25\bin\javac HelloWorld.java** (para Windows) se quiser continuar sem configurar as variáveis de ambiente.

Solução: Sugiro que você reveja o passo 9 do item 1.2 (para windows) ou passo 7 do item 1.3 (para Linux), do exercício 1.

4. Certifique-se de que o arquivo **HelloWorld.class** foi criado. O arquivo **HelloWorld.class** contém a representação **bytecode** da classe **HelloWorld**.

> **dir HelloWorld.class** (para windows)

ou

> **dir HelloWorld.class** (para linux)

6. Rode o programa HelloWorld usando o comando **java**. O comando **java** inicia a **Máquina Virtual Java** e executa o programa **HelloWorld** deste exemplo. Um programa em java pode ter múltiplas classes e um conjunto de bibliotecas de classes. Neste exemplo, o programa HelloWorld possui uma única classe chamada **HelloWorld.class**. O comando **java** é o **Interpretador Java**.

```
>java HelloWorld
HelloWorld world
```

Problema #2: Se você receber a mensagem de erro

```
>java HelloWorld
Exception in thread "main" java.lang.UnsupportedClassVersionError: HelloWorld
(Unsupported major.minor version 49.0)
```

É possível que você tenha compilado seu programa com uma versão diferente da que esteja tentando executá-lo.

Solução: certifique que sua variável de ambiente **PATH** esteja corretamente configurada para o diretório de instalação da versão apropriada do seu **JDK**.

Sugiro que você reveja o passo 9 do item 1.2 (para windows) ou passo 7 do item 1.3 (para Linux), do exercício 1.

Problema #3: Se você receber a mensagem de erro

```
>java HelloWorld
Exception in thread "main" java.lang.NoClassDefFoundError: HelloWorld
```

Provavelmente sua variável de ambiente **CLASSPATH** não está corretamente configurada com o diretório corrente.

Solução: tente usar o comando **java -classpath . HelloWorld** (Há um ponto “.” após o “-classpath”). Você verá mais sobre a classpath posteriormente.

```
>java -classpath . HelloWorld
```

7. Modifique **HelloWorld.java** como mostra a **Listagem-0.2** abaixo.

```
public class HelloWorld {
    /**
     * Comentário:
     * Primeiro programa em Java
     */
    public static void main( String[] args ){
        // também é um comentario
        // Imprime cadeia de caracteres "Hello world" na tela
        System.out.println("Olá Java");
        //modifique para imprimir mais
        System.out.println("Meu primeiro programa Java");
    }
}
```

Listagem-0.2: HelloWorld.java modificado

8. Compile e execute o programa. Observe a nova mensagem mostrada.

```
>javac HelloWorld.java
>java HelloWorld
```

Olá Java

Meu primeiro programa Java

2.2 Usar o compilador Java, comando *javac*

Neste passo você aprenderá como usar algumas opções do compilador **javac**.

1. Mostrar as opções do compilador **javac** usando a opção **-help**:

```
> javac -help
```

Usage: javac <options> <source files>

where possible options include:

- g* Generate all debugging info
- g:none* Generate no debugging info
- g:{lines,vars,source}* Generate only some debugging info
- nowarn* Generate no warnings
- verbose* Output messages about what the compiler is doing
- deprecation* Output source locations where deprecated APIs are used
- classpath <path>* Specify where to find user class files
- cp <path>* Specify where to find user class files
- sourcepath <path>* Specify where to find input source files
- bootclasspath <path>* Override location of bootstrap class files
- extdirs <dirs>* Override location of installed extensions
- endorseddirs <dirs>* Override location of endorsed standards path
- d <directory>* Specify where to place generated class files
- encoding <encoding>* Specify character encoding used by source files
- source <release>* Provide source compatibility with specified release
- target <release>* Generate class files for specific VM version
- version* Version information
- help* Print a synopsis of standard options
- X* Print a synopsis of nonstandard options
- J<flag>* Pass <flag> directly to the runtime system

2. Execute o comando **javac** com a opção **-verbose**

```
>javac -verbose HelloWorld.java
[parsing started HelloWorld.java]
[parsing completed 516ms]
[search path for source files: [.]]
[search path for class files: [C:\Program
Files\Java\jdk1.7.0_25\jre\lib\rt.jar,C:\Program Files\Java\jdk1.7.0_25\jre\lib\jsse.jar,
C:\Program Files\Java\jdk1.7.0_25\jre\lib\jce.jar, C:\Program
Files\Java\jdk1.7.0_25\jre\lib\charsets.jar,C:\Program
Files\Java\jdk1.7.0_25\jre\lib\ext\dnsns.jar, C:\Program
Files\Java\jdk1.7.0_25\jre\lib\ext\localedata.jar, C:\Program
Files\Java\jdk1.7.0_25\jre\lib\ext\sunjce_provider.jar, C:\Program
Files\Java\jdk1.7.0_25\jre\lib\ext\sunpkcs11.jar, .]]
[loading C:\Program Files\Java\jdk1.7.0_25\jre\lib\rt.jar(java/lang/Object.class)]
[loading C:\Program Files\Java\jdk1.7.0_25\jre\lib\rt.jar(java/lang/String.class)]
[checking HelloWorld]
[loading C:\Program Files\Java\jdk1.7.0_25\jre\lib\rt.jar(java/lang/System.class)]
[loading C:\Program Files\Java\jdk1.7.0_25\jre\lib\rt.jar(java/io/PrintStream.class)]
[loading C:\Program
Files\Java\jdk1.7.0_25\jre\lib\rt.jar(java/io/FilterOutputStream.class)]
```

```
[loading C:\Program Files\Java\jdk1.7.0_25\jre\lib\rt.jar(java/io/OutputStream.class)]  
[wrote HelloWorld.class]  
[total 1141ms]
```

3. Experiência com erros de compilação.

- Modifique o programa **HelloWorld.java** como mostra a **Listagem-0-3** abaixo. A mudança causa um erro de sintaxe – a palavra-chave **static** está incorretamente digitada como **statict**.

```
public class HelloWorld {  
  
    /**  
    * Comentário:  
    * Primeiro programa em Java  
    */  
    public statict void main( String[] args ){  
        // também é um comentario  
        // Imprime cadeia de caracteres na tela  
        System.out.println("Meu primeiro programa Java");  
    }  
}
```

Listagem -0.3: Código contendo erro de sintaxe

- Compile HelloWorld.java com o comando javac.
- Observe os erros

```
>javac HelloWorld.java  
HelloWorld.java:6: <identifier> expected  
    public statict void main( String[] args ){  
        ^  
HelloWorld.java:12: ';' expected  
    }  
    ^  
2 errors
```

- Remova o erro de sintaxe que você introduziu no código e recompile-o.

2.3 Usar o interpretador Java, comando *java*

Neste passo você aprenderá como usar algumas opções do interpretador **java**. Você pode pensar no comando como sua **Java Virtual Machine**.

1. Mostrar o uso das informações de ajuda do comando **java** com a opção **-help** ou **-?**.

```
>java -help  
Usage: java [-options] class [args...]  
    (to execute a class)  
    or java [-options] -jar jarfile [args...]  
    (to execute a jar file)
```

where options include:
-client to select the "client" VM
-server to select the "server" VM

-hotspot is a synonym for the "client" VM [deprecated]
The default VM is client.

-cp <class search path of directories and zip/jar files>
-classpath <class search path of directories and zip/jar files>
A ; separated list of directories, JAR archives,
and ZIP archives to search for class files.

-D<name>=<value>
set a system property

-verbose[:class/gc/jni]
enable verbose output

-version print product version and exit

-version:<value>
require the specified version to run

-showversion print product version and continue

-jre-restrict-search | **-jre-no-restrict-search**
include/exclude user private JREs in the version search

-? -help print this help message

-X print help on non-standard options

-ea[:<packagename>.../:<classname>]
-enableassertions[:<packagename>.../:<classname>]
enable assertions

-da[:<packagename>.../:<classname>]
-disableassertions[:<packagename>.../:<classname>]
disable assertions

-esa | **-enablesystemassertions**
enable system assertions

-dsa | **-disablesystemassertions**
disable system assertions

-agentlib:<libname>[=<options>]
load native agent library <libname>, e.g. **-agentlib:hprof**
see also, **-agentlib:jdwp=help** and **-agentlib:hprof=help**

-agentpath:<pathname>[=<options>]
load native agent library by full pathname

-javaagent:<jarpath>[=<options>]
load Java programming language agent, see **java.lang.instrument**

2. Execute o comando **java** com a opção **-verbose**

```
>java -verbose HelloWorld
[Opened C:\Program Files\Java\jre1.7.0_25\lib\rt.jar]
[Opened C:\Program Files\Java\jre1.7.0_25\lib\jsse.jar]
[Opened C:\Program Files\Java\jre1.7.0_25\lib\jce.jar]
[Opened C:\Program Files\Java\jre1.7.0_25\lib\charsets.jar]
[Loaded java.lang.Object from shared objects file]
[Loaded java.io.Serializable from shared objects file]
[Loaded java.lang.Comparable from shared objects file]
[Loaded java.lang.CharSequence from shared objects file]
[Loaded java.lang.String from shared objects file]
[Loaded java.lang.reflect.GenericDeclaration from shared objects file]
[Loaded java.lang.reflect.Type from shared objects file]
[Loaded java.lang.reflect.AnnotatedElement from shared objects file]
[Loaded java.lang.Class from shared objects file]
[Loaded java.lang.Cloneable from shared objects file]
[Loaded java.lang.ClassLoader from shared objects file]
[Loaded java.lang.System from shared objects file]
[Loaded java.lang.Throwable from shared objects file]
[Loaded java.lang.Error from shared objects file]
```

```
[Loaded java.lang.ThreadDeath from shared objects file]
[Loaded java.lang.Exception from shared objects file]
[Loaded java.lang.RuntimeException from shared objects file]
[Loaded java.security.ProtectionDomain from shared objects file]
[Loaded java.security.AccessControlContext from shared objects file]
[Loaded java.lang.ClassNotFoundException from shared objects file]
[Loaded java.lang.LinkageError from shared objects file]
[Loaded java.lang.NoClassDefFoundError from shared objects file]
[Loaded java.lang.ClassCastException from shared objects file]
[Loaded java.lang.ArrayStoreException from shared objects file]
[Loaded java.lang.VirtualMachineError from shared objects file]
....
[Loaded sun.net.www.URLConnection from shared objects file]
[Loaded sun.net.www.protocol.file.FileURLConnection from shared objects file]
[Loaded java.net.ContentHandler from shared objects file]
[Loaded java.net.UnknownContentHandler from shared objects file]
[Loaded sun.net.www.MessageHeader from shared objects file]
[Loaded java.io.FilePermission from shared objects file]
[Loaded java.io.FilePermission$1 from shared objects file]
[Loaded java.io.FilePermissionCollection from shared objects file]
[Loaded java.security.AllPermission from shared objects file]
[Loaded java.security.UnresolvedPermission from shared objects file]
[Loaded java.security.BasicPermissionCollection from shared objects file]
[Loaded java.security.Principal from shared objects file]
[Loaded java.security.cert.Certificate from shared objects file]
[Loaded HelloWorld from file:/C:/myjavaprograms/]
This is my first Java program!
[Loaded java.lang.Shutdown from shared objects file]
[Loaded java.lang.Shutdown$Lock from shared objects file]
```

3. Execute o comando **java** com a opção **-version**. Observe que o número da versão do JDK instalada é mostrado.

```
>java -version
java version "1.7.0_25"
Java(TM) SE Runtime Environment (build 1.7.0_25-b17)
Java HotSpot(TM) 64-Bit Client VM (build 23.25-b01 , mixed mode)
```